

OpenInfobutton Development Environment Setup Guide

Table of Contents

[1 Software Requirements 1](#_Toc365974898)

[2 Git and Maven 1](#_Toc365974899)

[2.1 Git 2](#_Toc365974900)

[2.2 Maven 8](#_Toc365974901)

[3 Building 17](#_Toc365974903)

[4 Running 18](#_Toc365974904)

# Software Requirements

* Java 7 JDK or better
* Tomcat 7
* Git SCM
* Maven 3.0.x
* Install a MySQL 5.0 server instance on the server you're building http://dev.mysql.com/downloads/mysql/
* JAVA\_HOME, CATALINA\_HOME, and M2\_HOME (Maven) environment variables all set
* PATH environment variable has the JAVA bin and the Tomcat bin directories
* Obtain a UTS account from here,

https://uts.nlm.nih.gov/home.html

# Git and Maven

Rather than using a specific IDE, this documentation will focus on retrieving and building the code from a command line. The specific instructions will be for a Unix shell, but it’s also possible to build from a Windows Command Line.

## Git

Git can be downloaded from <http://git-scm.com/downloads> , get the version for your specific platform. If on Linux, it’s a good idea to use a package manager like Yum (for Redhat) or AptGet (for Ubuntu) to install Git. Once installed and added to your path, use the following command to retrieve the source tree. Execute the command from the location on your file system you would like the code to reside.  
  
**https://github.com/VHAINNOVATIONS/Innovation-182.git**

You should now have a copy of the entire source tree on your local machine.

## Maven

The next step is building the OIB request service using Maven. You shouldn’t have to make any custom changes to your Maven settings.xml , with the possible exception of a proxy configuration, if you connect to the internet via a proxy...

**<proxies>**

**<proxy>**

**<active>true</active>**

**<protocol>http</protocol>**

**<host>**[**proxy.somewhere.com**](http://proxy.somewhere.com/)**</host>**

**<port>8080</port>**

**<username>proxyuser</username>**

**<password>somepassword</password>**

**<nonProxyHosts>**[**www.google.com**](http://www.google.com/)**|\*.**[**somewhere.com**](http://somewhere.com/)**</nonProxyHosts>**

**</proxy>**

**</proxies>**

**..**

Make sure Maven is added to your path and can be executed from a command line.

# Building

The build process has been simplified so that the individual modules are built in the proper order from a single POM. Navigate to the following directory on your command line,  
  
**Innovation-182/oib-request/**

Then simply execute the following Maven commands,  
  
**mvn clean**

**mvn –Dmaven.test.skip=true install**

The application should build without running unit tests. In order to run the unit tests, it’s necessary to install and setup a local instance of MySQL. I would recommend waiting until you’ve imported the project into an IDE of your choice and ensured the application builds without compilation errors first. If you want to run the unit tests at build time, after installing MySQL, you must the edit the following file,  
  
**Innovation-182\oib-request\oib-request-service\src\test\resources\serviceParams.properties**   
  
so the datasource parameters match your local database configuration.

Execute all the SQL scripts found in the Innovations-182/DeploymentPackage /sqlWithInsert directory.

Additionally, exectute the SQL script found in Innovations-182/sqlDump/DeploymentPackage/prodoib\_logs.sql.

# Running

The final step is to setup your knowledge resource profiles and terminology inference profiles directory. Knowledge resource profiles consist of the OpenInfobutton knowledge base. These profiles contain instructions that configure the different resources that OpenInfobutton will access to. These profiles will be taken from the database. The terminology inferences will be accessed from the specified file system location.

Check the database connection parameter strings found in the following properties files,

**oib-request-service/src/main/webapp/WEB-INF/serviceParams.properties**

Ensure they match the connection parameters configured for your local server. In the same file, set the parameter strings for an active UTS account.

You might also want to edit,

**Innovation-182/oib-request /oib-request-service /src/main/webapp/OpenInfobuttonDemo.html**

In that file edit the baseURL inside the callIM function (which points to University of Utah) to your own host.

Now the application can run. Since you should have already built all module dependencies in the previous step, it’s only necessary to rebuild **oib-request-service** after making the required changes to **serviceParams.properties** . Navigate to,  
  
**Innovation-182/oib-request/oib-request-service /**

and execute,  
  
**mvn –Dmaven.test.skip=true install**

This will generate the final WAR that you can copy to your Tomcat webapps directory. Copy it from,  
  
**Innovation-182/oib-request/oib-request-service/target/infobutton-service.war**to your Tomcat webapps directory and start Tomcat.